ARTICLE TEMPLATE

Deep learning-based detection of structural damage using time-series data

Hung V. Dang[a,e], Raza Mohsin[b], Tung V. Nguyen[c], T. Bui-Tien[d] and Huan X. Nguyen[a]

[a] Faculty of Science and Technology, Middlesex University, London, UK; [b] Department of Computer Science, Edge Hill University, UK; [c] Modeling Simulation Team, Schlumberger, Clamart, France; [d] Department of Bridge and Tunnel Engineering, Faculty of Civil Engineering, University of Transport and Communications, Hanoi, Vietnam; [e] Faculty of Building and Industrial Construction, National University of Civil Engineering, Hanoi, Vietnam

ARTICLE HISTORY
Compiled August 20, 2020

ABSTRACT
Previously, it was nearly impossible to use raw time series sensory signals for structural health monitoring due to the inherent high dimensionality of measured data. However, recent developments in deep learning techniques have overcome the need of complex preprocessing in time series data. This study extends the applicability of four prominent deep learning algorithms: Multi-Layer Perceptron, Long Short Term Memory network, 1D Convolutional Neural Network, and Convolutional Neural Network to structural damage detection using raw data. Three structures ranging from relatively small structures to considerably large structures are extensively investigated, i.e., 1D continuous beam under random excitation, a 2D steel frame subjected to earthquake ground motion, and a 3D stayed-cable bridge under vehicular loads. In addition, a modulated workflow is designed to ease the switch of different DL algorithms and the fusion of data from sensors. The results provide a more insightful picture of the applicability of Deep Learning algorithms in performing structural damage detection via quantitative evaluations of detection accuracy, time complexity, and required data storage in multi-damage scenarios. Moreover, these results emphasize the high reliability of 2DCNN, as well as the good balance between accuracy and complexity of Long Short Term Memory and 1D Convolutional Neural Network.

KEYWORDS
Vibration; machine learning; finite element methods; signal processing; structural health monitoring; neural network; time-series; structural analysis;

1. Introduction

Civil structures, whether they are buildings, bridges, dams, monuments, offshore platforms, nuclear build-outs, have been constantly subjected to unpredictable and rather violent excitations during their lifetime. Vehicular loads, wind forces, environmental changes, impact loads, and earthquakes can compromise structure integrity, making it vulnerable to any future harsh conditions and increasing chances of total collapse.

CONTACT A. N. Author. Email: D.viethung@mdx.ac.uk

Therefore, it is necessary to systematically monitor operational states to early recognize structural deficiencies, prevent catastrophic failures, and optimize maintenance strategies. Structural Health Monitoring (SHM) is achieved using a network of sensors installed across structures for measurement and assessment of current condition of structure under observation. With the development of sensor technology such as Wireless Sensor Networks (WSNs), Internet of Things (IoT), the measurement data can be collected in regular intervals to ensure safety of involved personnel and sustainability of the structure. However, turning massive measurement data into actionable information is a challenging task for the engineering community not only because of the high dimensionality of data but also inevitable uncertainties such as computational errors, signal noise, environmental effects.

Recently, deep learning (DL) techniques have gained considerable prominence. In comparison with conventional machine learning methods, Deep Learning possesses two superior properties: feature learning and scalability. The performance of the former depends substantially on handcraft features requiring domain knowledge and additional preprocessing data, while the latter is able to automatically extract meaningful features from raw data thanks to its hierarchical multi-layer architecture where lower layers identify basic features, and deeper layers synthesize higher-level patterns in terms of learned lower-level ones (Brownlee, 2019). Moreover, it is acknowledged that bigger data resort to larger architectures to maximize benefits and vice versa. That is why this artificial intelligent paradigm has become an exciting research direction, achieving impressive results in diverse research areas such as object recognition, machine translation, cancer detection, fault diagnostics, and so forth. The reasons for the success of the Deep Learning algorithms can be attributed to: 1) the development of high-performance computers with Graphic Processing Units (GPUs) and Tensor Processing Units (TPUs), 2) the availability of huge datasets due to the increasing digitalization of society (Goodfellow, Bengio, & Courville, 2016).

For SHM applications, deep learning-based methods offer promising tools to achieve end-to-end system as it does not require any additional signal preprocessing steps. Moreover, the DL models are applicable for various tasks such as damage detection, damage localization, damage severity, etc. by fine-tuning a few last layers of the architecture or using transfer learning techniques. Therefore, researchers in the engineering community have recently applied different neural networks models for SHM, such as multi-layer perceptron (MLP), Recurrent Neural Networks (RNN), Convolution Neural Networks (CNN).

A MLP consists of an input layer, single or multiple hidden layers, and an output layer. The number of hidden layers and the number of neurons per layer are hyperparameters of the network, usually determined via some search strategies such as grid search or random search. To detect faults in rotary machinery components, Lu, Wang, and Zhou (2017) developed a stacked autoencoder network composed of three hidden layers, providing direct health state classification from raw time series dataset without any prior signal processing techniques. Being interested in fault diagnosis of rotating machinery, Ma, Sun, and Chen (14) built a novel coupling deep neural networks which could capture joint information among multimodal sensory data to achieve high accurate prediction results. Galloway, Catterson, Fay, Robb, and Love (2016) studied fault diagnosis of tidal turbines by implementing a deep network. It was found that their deep learning approach achieved better results compared to feature-based methods involving k-nearest neighbor classifier, support vector machine, and decision trees.

Another deep learning architecture for handling time-series data is one-dimensional Convolutional Neural Network (1DCNN), which applies the convolution operator on

sequences of data points to extract informative features. Abdeljaber et al. (2018) proposed a 1DCNN model to detect changes in structural properties of a steel frame using measured acceleration signals. J. Li, Dackermann, Xu, and Samali (2011) published promising results for structural damage detection of Euler-Bernoulli beams by combining 1DCNN and original waveform signals in lieu of handcrafted features. Avci, Abdeljaber, Kiranyaz, Hussein, and Inman (2018) addressed the loss of connection stiffness of a steel frame structure via a novel SHM method using 1DCNN and WSNs. Zhang, Miyamori, Mikami, and Saito (2019) developed a 1D-CNN method for vibration SHM (VSHM) of bridge structures and successfully tested on both a simplified laboratory model and a real steel bridge. Ince (2019) demonstrated that the 1DCNN architecture was highly effective in real-time monitoring of motor conditions because their model took only 1.0 ms per classification, and the experimental accuracy was more than 97%. Recently, Liu and Zhang (2020) applied the CNN model to investigate the Maryland and Delaware highway bridges using historical data and achieves a high accuracy of more than 85%.

The recurrent neural network is a special architecture among Deep learning algorithms designed for capturing time-dependent characteristics. Thus, RNNs are naturally proposed for feature learning of sensor measurements. However, the sensor data usually consist of long sequential samples, therefore, the vanilla RNN suffers either gradient exploding or vanishing. To cope with this long-range dependencies, some derived RNN architectures such as Long Short Term Memory (LSTM) and its simplified version Gated Recurrent Unit were developed by scientists. Zhao, Wang, Yan, and Mao (2016) developed two LSTM-based methods, namely basic LSTMs, and Deep LSTMs, for SHM of high-speed CNC machines using sensory data. Their results confirmed that LSTM network could perform much better than a number of baseline methods. Yuan, Wu, and Lin (2016) investigated the remaining useful life of aero-engine utilizing LSTM under various operation modes and several degradation scenarios. They found that the standard version of LSTM itself has a strong ability to achieve accurate (both long term and short term) prediction during the degradation process. Lei, Liu, and Jiang (2019) developed LSTM-based method for fault diagnosis of wind turbines based on multi-sensor time-series signals. In their study, LSTM achieved best performance among neural network architectures, including the vanilla RNN, MLP, and Deep CNN.

The 2DCNN is a well-known architecture in computer vision which helps to build several deep learning applications surpassing human experts in image classification and object recognition. In order to leverage the advances of 2DCNNs, researchers have developed various techniques to convert one-dimensional sensory data into two-dimensional images, before feeding to a 2DCNN model, designed for damage detection tasks. Wang and Gao (2017) tackled the fault classification of the gearbox with the help of a 2DCNN-based method and time-scale images received from wavelet transformations of vibration signals. The proposed method could attain an accuracy of more than 99% in classifying different fault types in the tested gears. Lu et al. (2017) investigated bearing fault diagnosis by means of a 2DCNN using 2D feature maps of monitoring data. In their study, the 1D time-series signal was rearranged into 2D maps via a matrix reconstruction method. The comparison experiment results are competitive and promising as classification accuracies were more than 90% on average, and variation ranges were less than 2%. Recently, Tang, Chen, Bao, and Li (2016) applied 2DCNN to the anomaly detection problem of a long-span cable-stayed bridge, utilizing 2D representation of accelerometer sensor signals. Their results suggest that 2DCNN approach fared better than the existing method in structural anomaly detection, and was scalable to include more signal data from multiple measurants.

A notable trend towards DL techniques can be seen in various application areas, but there are still limited studies on the applicability of DL in structural health monitoring, especially for full-scale structures. Moreover, reviewed works only focused on one specific DL algorithm, but no extensive study comparing different DL algorithms in the civil engineering field on multi-facets, including detection accuracy, time complexity, and required data storage. This study is thus motivated by the need for a more insightful and complete picture of the applicability of DL algorithms to the Structural Damage Detection (SDD). The main contributions of this work are summarized as follows:

- Design of a modulated workflow flexible in switching different DL algorithms and fusing data from multiple sensors, thus enabling to 1) compare the practical effectiveness of various DL algorithms, 2) perform multiple structural damage detection tasks such as damage localization, damage severity, and 3) handle time series data polluted by noises through the noise injection learning method.
- Evaluation of strengths and weaknesses of each technique through three case studies of structures ranging from relatively small structures (i.e., a reinforced concrete beam) to considerably large structures (i.e. a stayed-cable bridge). It was found that 2DCNN constantly achieves the highest accuracy prediction, but it requires additional storage for images, while LSTM and 1DCNN demonstrate a fair balance between accuracy and computation complexity.

The rest of the paper is organized as follows: Section 2 briefs basic vibration signal analysis, including its time-domain representation, frequency-domain representation, and 2D time-frequency representation. Section 3 presents the architectures of four investigated algorithms: MLPs, LSTM, 1DCNN, and 2DCNN. In Section 4, the performance of these algorithms is compared through three case studies involving a 1D continuous beam, 2D steel frame, and 3D stayed-cable bridge. Finally, the conclusions are drawn in Section 5.

2. Deep Learning-based Structural Damage Detection

Currently, there is a vast number of techniques developed to improve the performance of DL, yet making these techniques practical and applicable to real scenarios is not a trivial task. One of the vital elements in building a successful DL application is to select an appropriate DL architecture, understand its impact on the application performance as well as its tradeoffs. Thereby, in this part, we investigate the applicability of the most widely used architectures across a spectrum of sectors (Jensen, Pedersen, & Thomsen, 2017) to SHM, namely, MLP, 1DCNN, LSTM, and 2DCNN after a brief introduction to the structural dynamic response.

2.1. Introduction to Structural Dynamic Response

In this work, acceleration signals serve as input for DL algorithms. A set of essential points across the structure is predefined; then, dynamic responses are measured and collected. It is common to apply a simple and rapid standardization processing on measured signals before feeding data into the DL model. The technique rescales data to a range having zero mean and unit variance. The mathematical formula of the

standardization technique is expressed as follows:

$$x_{new} = \frac{x - \mu}{\sigma}, \tag{1}$$

where x denotes the original data, x_{new} is the standardized data, μ and σ are the mean value and the standard deviation of the original data, respectively.

The time series sensory signal can also be represented in frequency domain via Fourier Transformation (FT). Any signal can be expressed as a sum of infinite series of sinusoids. The FT will decompose a signal into its constituent frequencies together with their associated coefficients. For civil structure, the frequency range of interest depends on both dynamic characteristics of structures and external excitations.

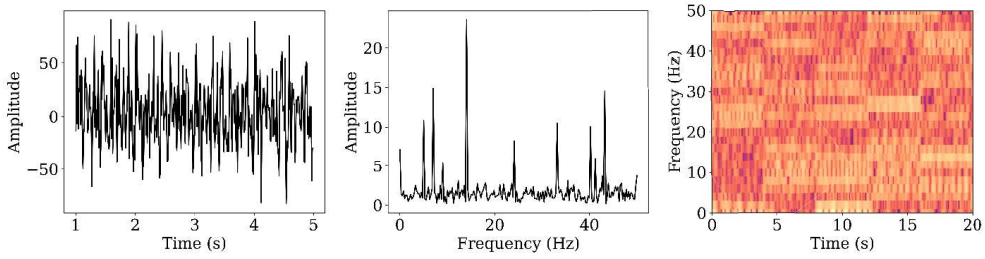
Moreover, it is possible to observe the evolution of frequency components over-time using the short-time Fourier Transform (STFT) technique, which results in a 2D representation of the signal. A long signal is divided into shorter segments of equal length; then FT is applied to each segment. Mathematically, the STFT is formulated as follows:

$$STFT(\tau, f) = \int_{-\infty}^{\infty} x(t)\psi(t-\tau)e^{-2\pi i f t}dt. \tag{2}$$

where $x(t)$ is the signal to be transformed, $\psi(t)$ is the window function, i is the imaginary unit, f and τ are frequency and time index of the STFT, respectively. Applying time discretization, the formulation is rewritten as follows:

$$STFT_{n,f} = \sum_{k=n}^{n+N-1} x_k \psi_{k-n} e^{-2\pi i f k}, \tag{3}$$

in which n stands for the segment number, N denotes the length of a segment. The 2D representation of STFT amplitude is called the spectrogram of the signal. Figure 1 illustrates an example of acceleration signal with its Fourier transformation and spectrogram.



(a) Time history. (b) Fourier transformation. (c) Spectrogram.

Figure 1.: Vibration signal a) time-domain representation, b) frequency-domain representation, c)Time-frequency representation.

2.2. Multi-Layer Perceptron-based Algorithm

The first investigated neural network is the multi-layer perceptron (MLP) composed of input layer, hidden layer and output layer. Measured signal data are fed into the input layer, which is then synthesized through one or multiple hidden layers before emerging at the output layer as damage diagnostic predictions. Mathematically, this process can be formulated as (Burkov, 2019):

$$Y = F(X|W) = f_L(...f_2(f_1(X|W_1)|W_2)W_L), \tag{4}$$

where L is the total number of layers in the network including the input and output layer, W is the matrix of parameters, X and Y denote the input and output vectors, respectively. The connection between two consecutive layers is performed by combining a linear matrix operation and a nonlinear activation function as follows:

$$f_l(X_l|W_l) = h(W_l \otimes X_l + b_l) \text{ with } l = 1, 2, ..., L \tag{5}$$

where $\otimes$ is the element-wise product, X_l is the input of the l-layer, b_l is bias vector, and h denotes the activation function that is a sigmoid function for hidden layers and the softmax function for the output layer. For this study, structural damage detection tasks are posed as multi-classification problems, e.g., detect damaged/undamaged state (0/1 class), assess damage severity (healthy, minor, medium, major class), etc.; thus the softmax function is selected at the output layer as a permanent part of the method. Softmax function normalizes evaluated values into a form of probability falling between [0, 1], probabilities of all classes are added up to 100%, and the class with the highest probability will be selected as predicted class. The softmax activation function is formulated as (François, 2017):

$$Y = f_L(X_L|W_L) = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_K \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{w_L^i X_L}} \begin{bmatrix} e^{w_L^1 X_L} \\ \vdots \\ e^{w_L^k X_L} \\ \vdots \\ e^{w_L^K X_L} \end{bmatrix}, \tag{6}$$

where K is the number of output classes, y_k is the predicted probability of the class k, w_L^k is the weight row vector k of the parameter matrix Θ_L associated with the k^{th} output class, i.e.: $W_L = [w_L^1 ... w_L^k ... w_L^K]^T$.

Herein, the network parameters are computed in a supervised fashion, which means the structural state is labeled in advance, next the database are divided into three groups, namely training, validation, and testing datasets. The value of network parameters are randomly initialized, then they are iteratively updated via the backpropagation algorithm to minimize the deviation between prediction values and annotated structural states. At a step t of the training process, the parameter values are updated by:

$$w_l^{ij}(t) = w_l^{ij}(t-1) + \eta \delta_l^i x_{(l-1)}^j(t), \tag{7}$$

where l denotes the layer number, δ_i^l is the error at node i of layer l, which is calculated backward from the deviation at output layer δ_L and η stands for the learning rate.

The training process terminates when a tolerable error level is met or the number of iterations reaches to a limit.

The deviation between the prediction and actual structural states is computed via the cross-entropy function (Janocha & Czarnecki, 2017) as follows:

$$Loss = \delta_L = -\sum_k y^k \log p(\hat{y}^k), \tag{8}$$

where k denotes the class number in output, p is the probability estimate, y is the actual label and $\hat{y}$ stands for prediction value.

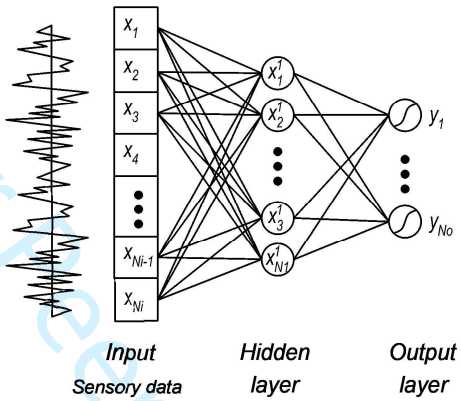


Figure 2.: Architecture of the Multi-Layer Perceptron Network dedicated for VSHM

Theoretically, MLP network with one hidden layer can approximate any functions in the training process on the condition of a sufficiently large hidden layer (Goodfellow et al., 2016). However, such a large hidden layer usually leads to the over-fitting phenomena (accuracy on test dataset is significantly lower than training accuracy) and requires a high computational cost. In this work, the number of output classes is basically under 100 therefore, number of perceptrons in hidden layer are set to 1000. Figure 2 shows the architecture of the implemented MLP with 1000-perceptron hidden layer.

2.3. Long Short Term Memory-based Algorithm

In this section, LSTM network is reviewed and adapted for structural damage detection purposes. The fundamental theory of LSTM refers to the work of (Hochreiter & Schmidhuber, 1997). The structure of LSTM consists of jointly connected repeating cells. Each cell has three gates, namely forget gate, input gate, and output gate for controlling information flow. Outputs of LSTM sequences are fed into a fully connected layer with softmax activation function, which further provides the probability for each predicted class (see Figure 3).

The mathematical formulation of this model is described as follows. Given x_t, the input at time step t, h_{t-1} the output of LSTM cell at time step $t-1$, a linear transformation of the combination of x_t and h_{t-1} is expressed by:

$$\mathcal{L}(h_{t-1}, x_t) = W[h_{t-1}, x_t] + b, \tag{9}$$

where W and b are the weight matrix and bias vector of the network.

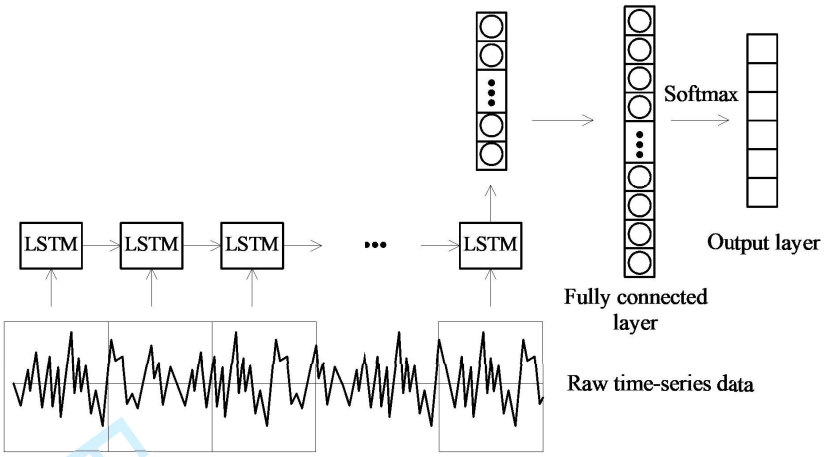


Figure 3.: Architecture of the Long Short Term Memory Neural Network (Lei et al., 2019)

Then, the three gates inside each cell of LSTM are written (Olah, 2015):

$$\begin{aligned} f_f &= \sigma(\mathcal{L}_f(h_{t-1}, x_t)) \\ f_i &= \sigma(\mathcal{L}_i(h_{t-1}, x_t)) \\ f_o &= \sigma(\mathcal{L}_o(h_{t-1}, x_t)), \end{aligned} \tag{10}$$

in which the subscripts f, i, o indicate the forget gate, input gate and output gate, respectively. σ is the sigmoid activation function.

At every time step t, the new candidate of information is firstly created by applying the tanh activation function on a linear transformation of the combination $[h_{t-1}; x_t]$:

$$C_t = tanh(\mathcal{L}_c(h_{t-1}, x_t)), \tag{11}$$

where $\mathcal{L}_c$ is a creation linear operator whose formula is similar to Eq. (9). Then the flow of information from previous step is updated with the new candidate by the gates of LSTM cell:

$$s_t = (f_f \odot h_{t-1}) \oplus (f_i \odot C_t), \tag{12}$$

where $\odot$ and $\oplus$ are element-wise product and element-wise addition. Next, the output of the cell at time step t is calculated based on the updated information and the output gate:

$$h_t = f_o \odot s_t. \tag{13}$$

2.4. 1D Convolution Neural Network-based Algorithm

It is commonly acknowledged that the convolutional neural networks (CNNs) can provide outstanding performance on signal classification and pattern recognition because 1) its architecture is especially suitable for discovering local relationships in space; 2) it reduces the number of network parameters, thus leading to a lower computational complexity compared to conventional neural networks. Once vibration data enter into

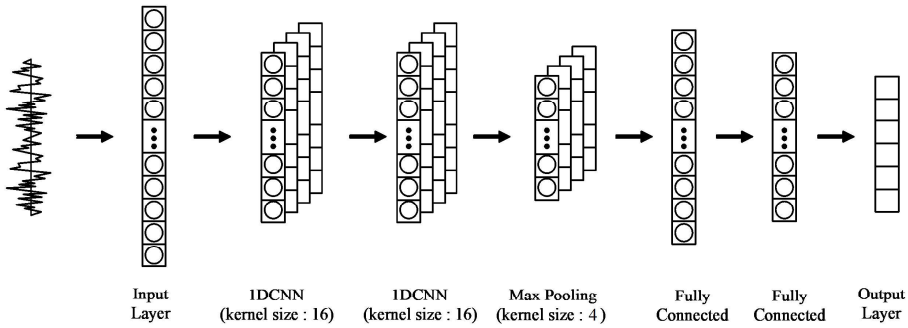


Figure 4.: Architecture of 1DCNN for structural damage detection adapted from (Lin et al., 2017)

the network, the 1DCNN layers will extract inner relationships between measured points and their higher derivatives before feeding to the last fully connected layers serving as a classifier. Figure 4 depicts the detailed configuration of the applied CNN for structural damage identification, which is developed from the work of (Lin et al., 2017). Its configuration included a sequence of layers involving an input layer, two convolutional layers, a max-pooling layer, two fully connected (FC) layers, and an output layer.

The hyperparameters of a 1D convolution layer comprise the number of kernels, the kernel length, and the stride value. The relation for one kernel is expressed as follows (Kiranyaz et al., 2019):

$$h_k = conv1D(w_k, X) + b_k, \tag{14}$$

where h_k, w_k and b_k are respectively the output vector, weight vector and bias parameter of the kernel k. X is the input vector, and $conv1D$ is the 1D convolution operator whose i^{th} output is calculated as follows:

$$conv1D(w_k, X)(i) = w_k * X(i) = \sum_{j=1}^{N_k} w_{kj} x_{i-j}, \tag{15}$$

where N_k is the length of the kernel k, w_{kj} is the j^{th} element of vector w_k.

In fact, this 1DCNN architecture can be extended to a deep CNN configuration involving multiple consecutive convolutional layer to perform complex tasks requiring various feature abstraction levels.

2.5. 2D Convolution Neural Network-based Algorithm

The fourth deep learning algorithm is based on the deep 2DCNN, which utilizes images instead of temporal signals as input data. The vibration signal is translated into the time-frequency representation through the short-time Fourier Transform technique, as presented in Section 2. Then, a powerful image classifier in the Artificial Intelligent literature is employed to detect abnormally structural cases. Here we adopted the ResNet-18 classifier and employed transfer learning technique to fine-tune the model for structural damage identification tasks. The ResNet-18 is chosen due to its competitive performance, and explicit feed-forward architecture which is depicted in Figure 5.

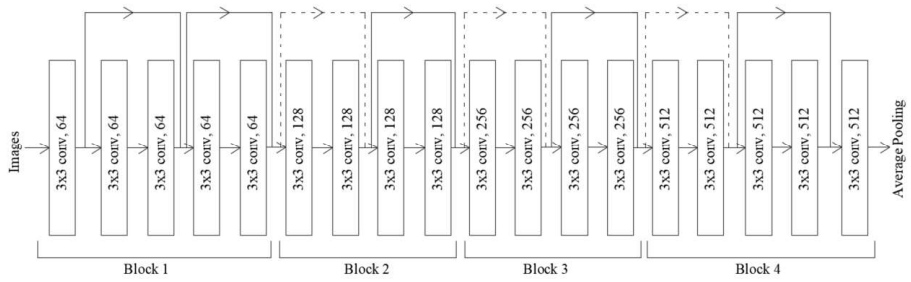


Figure 5.: Architecture of the ResNet-18 Network (He et al., 2016)

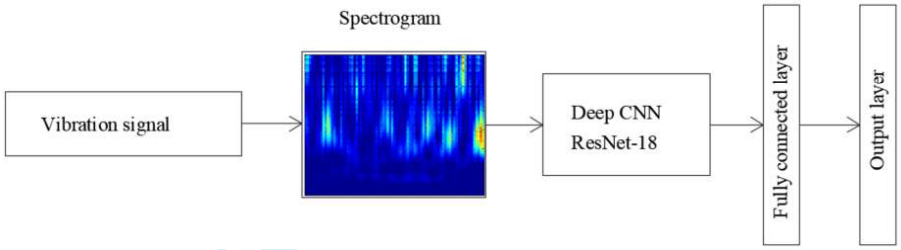


Figure 6.: Workflow of 2DCNN-based method for structural damage detection

There are in total 18 layers of 2D convolution neural layers divided into four blocks, and shortcut connections are inserted between every two CNN layers (He et al., 2016).

Besides, the transfer learning technique allows exploiting general knowledge pre-trained with a vast amount of data of different categories in a particular domain. For further details of the transfer learning technique, refer to (Pan & Yang, 2009) for its widely cited unified definition and recent works of (Howard & Gugger, 2020) for its practical implementation API. The reasons of using this technique beforehand are two folds: 1) the total number of parameters of ResNet-18 is more than 11 million, thus training the whole model from scratch requires impractically large volumes of data and time. 2) There exists a broad consensus that for computer vision, features extracted from base models well-performing on general tasks benefit the performance of target models in specific domains because features are more generic in early layers and more task-specific in later layers (Lee, Karpathy, & Johnson, 2018). Herein, the parameters of the ResNet-18 model are pretrained with millions of images from the ImageNet dataset from Google.

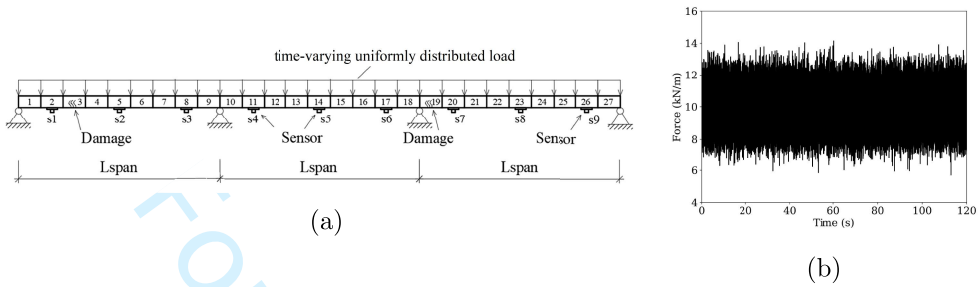
The workflow of the algorithm is illustrated in Figure 6. Firstly, original vibration signals are fed into a preprocess module to be converted into images of time-frequency representation, prior to entering the ResNet-18 models. Afterwards, computed data from the ResNet is fed into a fully connected layer before giving the classification results at the output layer using the softmax activation function.

3. Performance Evaluation

In this section, the implementation of the DL algorithms is investigated through three case studies from a simple 1D structure to a full-scale 3D structure. Their performances in terms of accuracy, time complexity, required storage as well as the robustness in dealing with noisy data are quantified and compared based on calculated results.

The next subsection will go through three case studies from a simple one-dimensional beam structure to a two-dimensional frame structure to a three-dimensional full-scale stayed-cable bridge.

3.1. Case Study 1: 1D Continuous Beam Excited by Random Force



(a) (b)

Figure 7.: Three-span continuous beam under time-varying uniformly distributed load. a) Schematic representation of the beam. b) Example of a time-varying amplitude loading at the middle of the beam

Table 1.: Numerical input parameters for the continuous beam case study.

Number of mode	f_1 (Hz)	f_{10} (Hz)	Duration of simulation (s)	Time step (s)	Mesh size (m)	Element type	Analysis method
10	11.67	169.67	120	1e-3	0.2	B21	Modal analysis

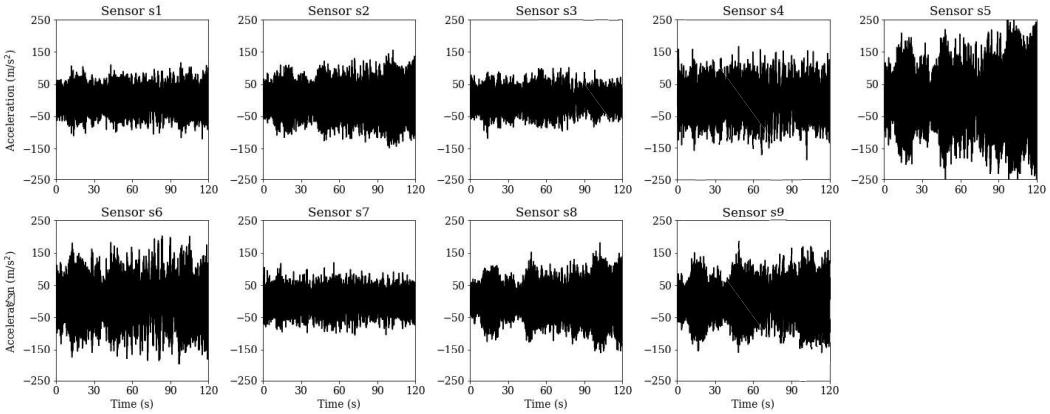


Figure 8.: Time history of acceleration recorded at virtual sensors from s1 to s9.

The first numerical experiment is a continuous concrete beam usually occurring in buildings subjected to time-varying uniformly distributed loads reflecting machinery excitation, for example. The beam consists of three spans of equal length of 9 m (Figure 7). Its cross-section dimensions are 40x60 cm^2, Young's modulus E is 30 GPa with 95% confidence interval being $E_{95} = (1 \pm 0.05)/E$ (Noguchi & Nemati, 1995), Poisson's ratio is 0.3, mass density ρ is between 2200 and 2400 kg/m^3 (Kosmatka, Kerkhoff, &

Panarese, 2019), and the modal ratio is 2%. The beam is modeled in the Finite Element software Abaqus Standard (Dassault, 2016a), using the three dimensional beam element. The element has six degrees of freedom (dofs) at each node, including three translations and three rotations around the x, y, and z directions. Section properties and material properties are summarized in Table 1. The beam is subjected to a time-varying uniform load modeled as a Gaussian random process with amplitude falling between 1 and 20 kN/m, and the coefficient of variation is 0.1. Figure 7(b) shows a representative time history of load amplitude at the middle of the beam. The self-weight is implicitly accounted for in the simulation through the given material mass density and gravitational acceleration.

Each span of the continuous beam is divided into nine segments of 1 m length. The damage is randomly assigned to these segments. The damage is modeled as a reduction in the effective height of the section, leading to element stiffness loss, thus resulting in changes in the dynamic behavior of the structure. Three damage levels are investigated, corresponding to 5%, 15%, and 25% reduction in the effective height. In reality, this type of damage is often observed in reinforced concrete elements prone to cracking, spalling, and so forth. The monitoring system is simulated by using 9 virtual accelerometers measuring vertical acceleration responses of the beam; three vrtual sensors are mounted on each span (at 1/4, 1/2, and 3/4 span). The setup configuration is schematically depicted in Figure 7. It is supposed that virtual sensors work in ideal conditions, which means there is no noise included in the obtained acceleration time-series. The variation of responses only results from uncertainties in input parameters and various damage scenarios introduced. The inputs for the Monte Carlo simulation are summarized in Table 2, other parameters remain identical between calculations. The setup configuration is schematically depicted in Figure 7.

Table 2.: Random parameters for Monte Carlo simulations of the continuous beam structure.

Parameter	E (kN/m2)	ρ (kg/m3)	Loading (kN/m)	Damage level*(%)	Damage location
Distribution	Normal	Uniform	Uniform	Uniform	uniform
Value	$E_{95} = [28.5 - 31.5]$	[2200 - 2400]	[1 - 20]	[0, 5, 15, 25]	[s1-s9]

*0 : *No damage.*

Due to the uncertainty of external load, material properties' variation, and various damage scenarios, we use the Monte Carlo method to generate an input database of 10000 samples and run extensive finite element simulations. The finite element simulation is realized by the use of the mode-based dynamic method in Abaqus Standard, which provides the time-dependent response of the structure via the principle of superposition. At first, the eigenvalue analysis procedure (subspace projection method) is conducted to extract modal characteristics, the number of extracted modes should be sufficient to reflect the real structural behavior closely. After that, the amplitude coefficient associated with each mode in response to external excitations is determined; the response of the structure is obtained by superposing modal responses. More details of the mode-based dynamic analysis can be found in (Dassault, 2016c). In general, modal dynamic procedure could provide results with satisfied accuracy but requires less computational effort than the direct integration, therefore it is suitable to generate an extensive dataset for DL algorithms. The number of modes included the modeling is selected through a conventional convergence analysis, resulting in a value of 10 for

the present simply supported beams. The simulation duration is T=120 s, and the sampling frequency is f=200 Hz. Figure 8 illustrates an example of time-series sensor data obtained from numerical simulations.

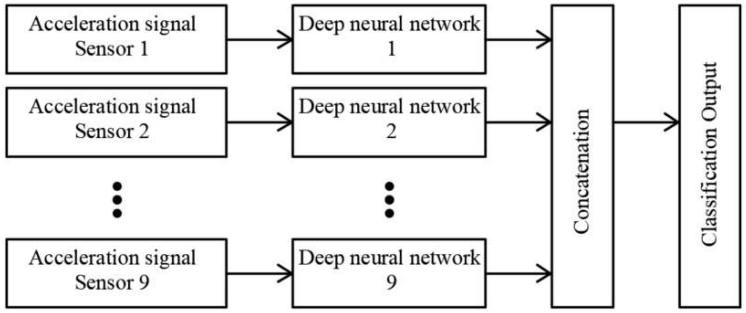


Figure 9.: Concatenating calculation outputs of different neural networks into final classification results. Acceleration signal retrieved by different virtual sensor is fed to the corresponding neural network.

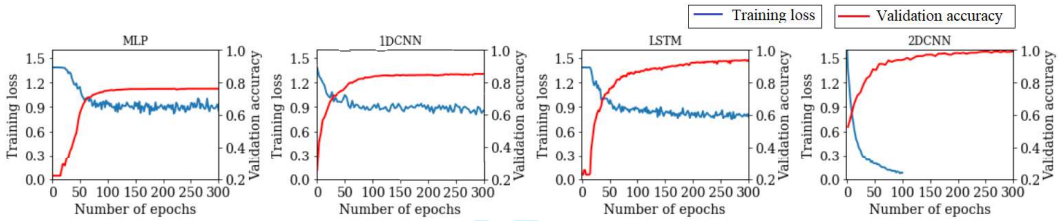


Figure 10.: Evolution of training loss and validation accuracy in function of the number of epochs in the training process.

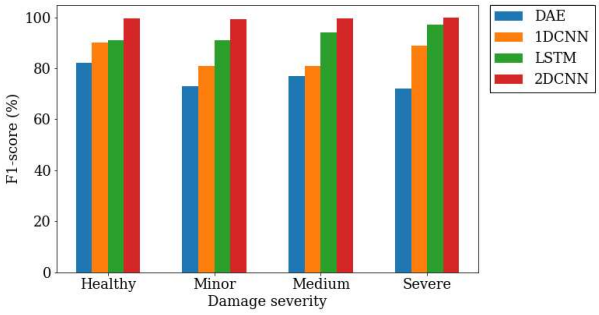


Figure 11.: Comparison of F1-score obtained by different DL algorithms on the test database for damage severity task.

Intuitively, using time-series from only one sensor hardly provides accurate results about damage localization across the whole structure, while taking multiple sensor data as input for the neural network model can benefit the model performance. There are two ways to handle multiple time series: i) combining these time series into a 3D time-amplitude-sensor tensor or even 4D time-frequency-amplitude-sensor tensor before feeding to a neural network ii) each time series enters to a separate model of the same architecture. The former way needs to increase the dimension of input data,

while the latter one requires to repeat training process by the number of networks. To cope with these issues, one revisits the second approach by concatenating the output of each neural network into a long vector before passing through a common output layer with the softmax activation function to provide damage classification output, as schematically illustrated in Figure 9. In this way, the resulting unified model has the capability of using the original multiple time series as input without increasing input dimensions or repeating the training process.

Table 3.: Categorization of damage severities.

Damage severity	No damage (Label "O")	Minor damage (Label "I")	Medium damage (Label "M")	Severe damage (Label "S")
Height reduction	∼0%	∼5%	∼10%	∼15%

The first structural damage detection task is to estimate damage severity. The data are labeled by the four introduced damage levels, namely, healthy, minor, medium, and major, as listed in Table 3. The damage classes are evenly distributed, i.e., there are 2500 cases per damage level. Such uniform distribution simplifies the training process by avoiding the class imbalance problem. Afterward, they are divided into three groups: training, validation, and testing with a ratio of 80/10/10 for training and testing the performance of presented DL algorithms. The training process aims to determine the network's parameters, which minimize the deviation between the predicted outputs and true labels measured herein by the Cross-Entropy loss function (Goodfellow et al., 2016) by means of the Adam optimizer algorithm (Kingma & Ba, 2014). The learning rate standing for the relative amount of DL model weights updated after each optimization iteration is set to $lr = 0.0001$, the batch size, i.e., the number of data utilized in one optimization iteration is 32. These values are commonly selected in Machine Learning practices and were found effective in this work.

Figure 10 shows the evolution of training loss and validation accuracy of four neural network algorithms obtained during the training process. Apparently, 2DCNN model achieves the best validation accuracy, followed by LSTM, IDCNN and MLP respectively. On the other hand, 1DCNN and MLP have fast convergence speeds, the curves follow a stable trend after around 100 epochs.

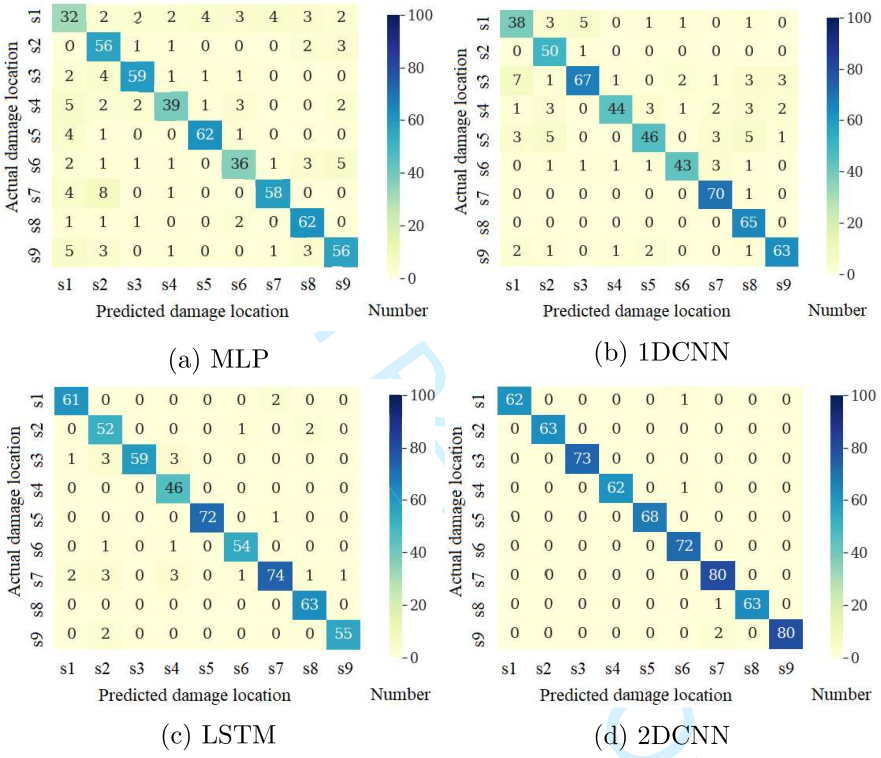
Given a specific structural class denoted C, True Positive (TP) is the number of outcomes where the model correctly predicts the class C. True Negative (TN) is obtained when a time series comes from another class, and the model correctly avoids assigning C for the time series. Inversely, False Positive (FP) is where the model mislabels an input as C while it is not, and False Negative (FN) is when the model ignores the true label C of the input. Then, the precision P, recall R and F_1 metric are derived as follows:

$$\begin{aligned} P &= \frac{TP}{TP+FP} \\ R &= \frac{TP}{TP+FN} \\ F_1 &= \frac{2}{1/P+1/R}. \end{aligned} \tag{16}$$

Figure 11 highlights the F1-score obtained with four deep learning algorithms on testing data. It is noted that all deep learning algorithms can discriminate between

healthy states and minor damaged states (5% section loss), which are visually difficult to recognize. Moreover, the best testing results are obtained with 2DCNN algorithms, followed by LSTM, then 1DCNN and MLP.

The second structural damage detection task is damage localization. In reality, structures consist of a huge number of elements, therefore, the sensors are strategically placed at essential elements or distributed uniformly across the structure, and damage localization tasks are focused on the elements associated with sensor placement. The process of the damage localization in this example is realized as follows. At first, the fictive damage is introduced to an element corresponding to sensor locations (9 possible cases), the dataset is further labeled by respective damage locations. The number of neurons in the output layer of the network is set to 9. The training process is then carried out similarly as damage severity task.



(a) MLP (b) 1DCNN

(c) LSTM (d) 2DCNN

Figure 12.: Confusion matrix for damage localization obtained by different DL algorithms.

Figure 12 details the testing results obtained with four deep learning algorithms represented via confusion matrices, in which the horizontal axis denotes classes predicted by models, the vertical axis represents the actual ones, and the number of each class is visualized by color intensity. The 2DCNN and LSTM provide highly accurate damage identification output. Although, some false predictions are observed, which are related mostly to the location s1 and s9, near two extremities of the beam.

Table 4 summarizes the main parameters of four algorithms, including training times, testing accuracies for both damage severity, and damage localization tasks. All the experiments are carried out on a high-end machine equipped with 6 CPU Intel Xeon, 32-GB RAM, and 2 Geforce GTX 1080Ti GPU.

Remark 1: The fusion DL workflow described above faciliates the detection of the

damage occurring across the whole structure, which is infeasible if using time series from a single sensor, especially when the sensor is far apart from damage locations. One also observed that 2DCNN achieves the best performance (99.2%) but requires the most computational time, i.e., 1.6 times higher than that of LSTM, which also yields good identification results (95%). The MLP and 1DCNN are notably suitable for a quick assessment of structural conditions due to its rapid implementation and convergence speed, whose training time is only one-third of that of 2DCNN.

Table 4.: Damage detection results obtained with different Deep learning algorithms.

Task	Damage severity				Damage localization			
Model	MLP	1DCNN	LSTM	2DCNN	MLP	1DCNN	LSTM	2DCNN
Number of epoch	300	300	300	100	300	300	300	100
Total traing time (s)	900	1188	2250	3650	712	807	1918	2100
Testing accuracy	78.8%	87.5%	93.0%	98.9%	83.9%	86.0%	95.0 %	99.2%

3.2. Case Study 2: Experimental dataset

In this case study, these investigated neural network architectures are applied to an experimentally measured vibration data from a three-story frame structure realized at *Los Alamos National Laboratory* (2002) as shown in Figure 13 with detail dimensions. The reasons for selecting this dataset are its validity, clarity, and particularly a sufficient number of available data for building deep learning applications. The frame consists of Unistrut columns, and aluminum plates with 1.3 cm thickness and joined together using bolt-to-bracket connections. The base of the frame is supported by four air mount isolators, enabling the structure to move horizontally.

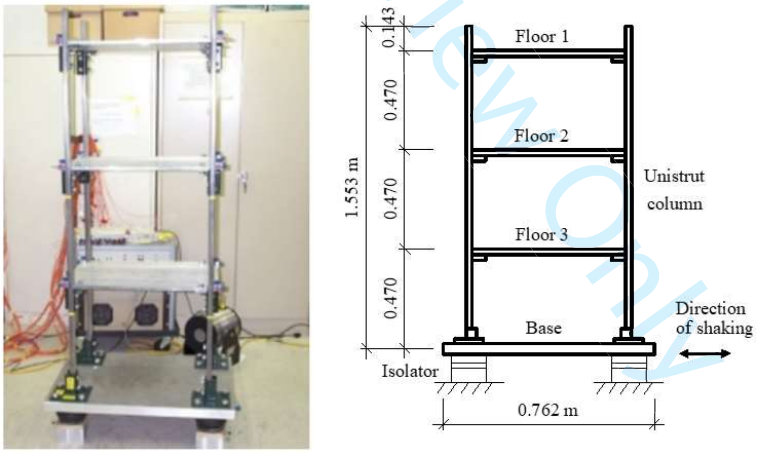


Figure 13.: Three-story frame structure experiment (*Los Alamos National Laboratory*, 2002)

The above setting of the structure is considered as the undamaged condition. Thereafter, various modifications are introduced to the structure to generate different damaged states, including loosening or completely removing column-plate connections at a corner of the first floor or third floor, or at both locations. Structural states and corresponding labels are enumerated in Table 5. There are 5 undamaged states, 3

Table 5.: Structural states of the three-story frame structure

State	1	2	3	4	5	6	7	8	9	10
Damage	0	0	0	0	0	1	1	1	1	1
Location	0	0	0	0	0	L1C	L1C	L1C	L1C	L1C
Level	0	0	0	0	0	DB0	DBB	D05	D10	DHT
State	11	12	13	14	15	16	17	18	19	20
Damage	1	1	1	1	1	1	1	1	1	1
Location	L3A	L3A	L3A	L3A	L3A	L13	L13	L13	L13	L13
Level	DB0	DBB	D05	D10	DHT	DB0	DBB	D05	D10	DHT

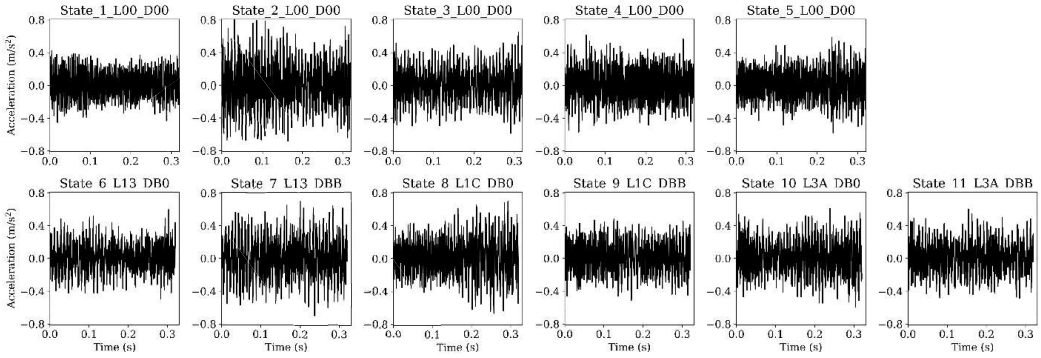


Figure 14.: Acceleration time series measured at the second floor for 11 structural states.

damages locations, and 2 damage levels, which result in total 11 structural states, where L1C, L3A, and L13 denote damage locations on the first floor, third floor, and at both floors, respectively. DB0 refers to the removal of the bolt, and DBB represents the bracket was completely removed.

The structure is excited through an electrodynamic shaker at the corner of the base floor, the vibration of each story is measured by using 8 single-axis accelerometers with 1 V/g sensitivity and a sampling frequency of 1600 Hz, two sensors installed for each column-plate joints (in x and y directions), resulting in total 24 accelerometers. Each structural state is measured multiple times with different levels of shaker excitation. For each sensor, there are in total 1080 time-series of length 2048 corresponding 11 structural states. Figure 14 shows a representative example of vibration signals from a sensor on floor 2. It is difficult to visually distinguish damaged states from undamaged ones. The vibration database is split into training/validation/testing datasets with a ratio of 80/10/10. Next, one adapts the present algorithms to perform three damage detection tasks, meaning identify undamaged/damaged state, damage localization, and damage severity. Input for neural network models is 24 time-series of length 2048, labeled by corresponding structural states as mentioned in Table 5.

Damage detection of investigated methods is presented in Table 6 and in Figure 15 in terms of efficiency and accuracy. As expected, the 2DCNN achieves the highest accuracy in all three tasks: 99% for damage detection, 95% for damage localization, and 94.6% for damage severity, following by LSTM, 1DCNN, and MLP. In the figure, the efficiency is estimated based on training time, required storage and number of trainable parameters; results are normalized between 0 and 1 for a better comparison, where higher value denotes a better efficiency. It is noticed that MLP has a number

Table 6.: Damage detection results on experimental database of a 3-story frame.

Task	MLP of trucks	1DCNN (m/s)	LSTM loss (%)	2DCNN location
Detection accuracy (%)	97	97	99	99
Localization accuracy (%)	87	93	93.5	95
Severity accuracy (%)	85	91	92.6	94.6
Training time (s)	350	470	930	1250
Number of trainable parameters	389806	342670	326606	540550
Storage (Mb)	1010	1010	1010	1945

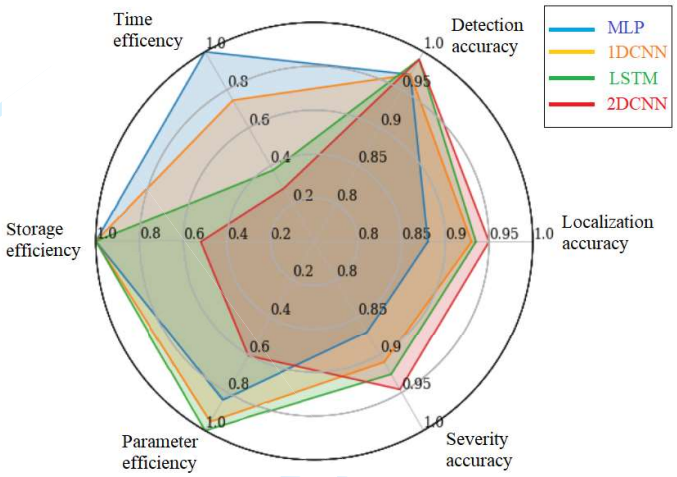


Figure 15.: Damage detection results in terms of accuracy and efficiency.

of trainable parameters close to those of 1DCNN and LSTM, but its architecture is highly suitable for parallel calculation then the training time is significantly lower than that of LSTM's recurrent architecture.

Remark 2: In summary, the validity of the present methods is successfully proved through experimental data from laboratory testing. Different SDD tasks can be performed by reassigning respective labels for time-series input, fine-tuning models' parameters, without the need of redesigning entire neural network architectures.

3.3. Case Study 3: 2D Frame Subjected to Seismic Motion

The second case study is a two-bay, five-story steel frame structure, as highlighted in Figure 16(a) with the dimension details and the element numbering (Fragiacomo, Amadio, & Macorini, 2004). The characteristics of the frame are as follows: the span length is 6 m, the story height is 3.5 m except for the ground story whose height is 4 m. The steel material has modulus of elasticity E=210 GPa, strength f_u = 360 Pa, mass density ρ = 7850 kg/m^3. The beams are made from European steel profiles: IPE400 for those of the first three floors, IPE360 for the fourth and fifth floors, HEB260 for outer columns, and HEB320 for middle columns. Damping is modelled by using the Rayleigh approach, the damping ratio for the vibration mode i is calculated as follows

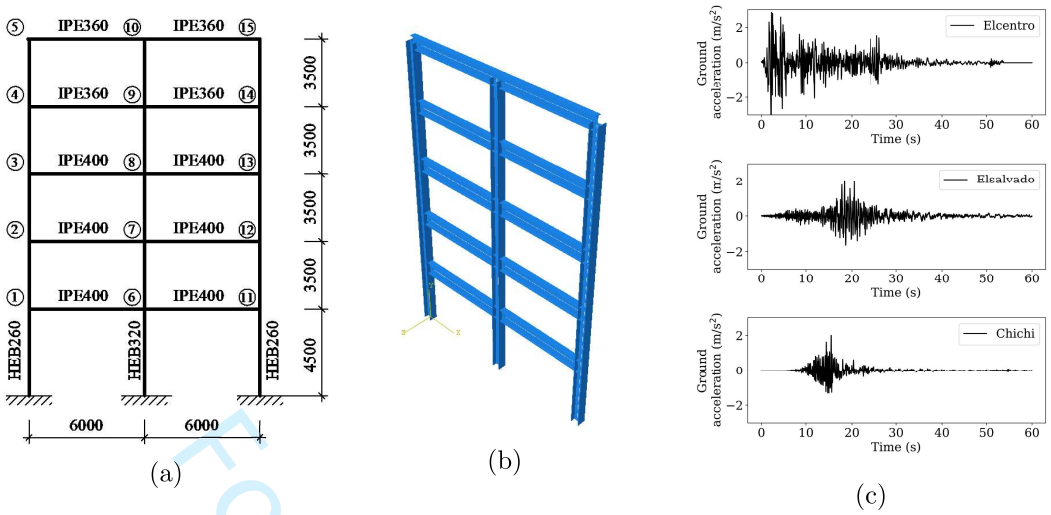


Figure 16.: a) Schematic representation of the two-bay five-story steel frame subjected to earthquake ground motion. b) 2D model of the steel frame in ABAQUS. c) Ground accelerations of three investigated earthquake: Elcentro, Elsalvado and Chichi, respectively (Haddadi et al., 2008).

(Cruz & Miranda, 2017):

$$\zeta_i = \frac{1}{2\omega_i} a_0 + \frac{\omega_i}{2} a_1. \tag{17}$$

where ω_i is the circular frequency of mode i, a_0, a_1 are two parameters to be determined. Setting a damping ratio $\zeta = 0.02$ for the first two modes ($f_1 = 3.36Hz, f_2 = 10.31$), results in the constants a_0 and a_1 to be 0.31828 and 0.000466, respectively.

The column-beam connection in frame structures is a critical element which significantly affects structure behavior. In this work, the connection is modeled as a semi-rigid link with rotational spring, which has been proven to reflect the dynamic characteristics of structures accurately (Csébfalvi, 2007). Value of rotational stiffness of the semi-rigid connection is estimated as follows (Sekulovic & Salatic, 2001):

$$c = \frac{3EI}{l} \left(\frac{\lambda}{1-\lambda} \right), \tag{18}$$

where c denotes the connection rotational stiffness, l is the length of the structure element, λ is a fixity factor ranged from 0 to 1, I is the moment of inertia of the element cross-section.

In fact, it is convenient to train a model able to detect underlying damages from free or ambient vibration measurements because of their accessibility. On the other hand, using vibration signal measured during and shortly after excitation is an appealing alternative because obtained signals are rich in information in terms of amplitude, variability, and frequency contents. Since resulting models can depend on specific excitation, e.g., earthquake records, it is beneficial to consider different earthquake records, enriching the diversity of database and robustness of DL methods. Specifically, for this case study, the frame is subjected to three simulated ground motion with characteristics of the 1940 El Centro earthquake, the 1999 Chi-Chi earthquake, and the 2001 El

Salvador earthquake. The ground acceleration time histories of these earthquakes are highlighted in Figure 16(c). Virtual accelerometer sensors were placed at the frame connections to capture frame movements during ground motions.

In general, damages do not take place before earthquake events, but during or shortly after. Therefore simulations are performed in two phases: The first phase corresponding to the first half 30 s of earthquake records where the structure is supposedly still intact, then in the second phase, connection stiffness reductions of different levels, i.e., 0% (healthy), 30%, 60%, and 90%, are arbitrarily introduced to two connections (multiple damage scenarios) by reducing rotational stiffness c via Eq. 18, other connections are kept unchanged. The selection of 30 s adopted here comes from the observation that most significantly strong part of EQ (Figure 16) records falls between [0-30 s], specifically the root mean square (RMS) value of the first 30 s is greater than 8 times that of the 30s second part for all three considered EQ records, thereby it is supposed that the second part does not incur further damage. The recorded 60 s-length acceleration is used as input data to train the deep learning models further. As changes in connection stiffness could lead to changes in global stiffness of the structure, using complete signals recorded from before to after damage event, could store property shift in a single data. Thus, it will be advantageous for classifiers, especially those having the capability of exploiting time-varying dependence such as LSTM and 2DCNN. A procedure using only a portion of signals recorded after the occurrence of damage is also feasible, but it will require corresponding undamaged data to identify property shifts, especially when working with different seismic spectrums triggering different eigenmodes of the structure.

Table 7.: Numerical input parameters for the 2D steel frame case study.

Number of mode	f1 (Hz)	f20 (Hz)	Duration of simulation (s)	Time step (s)	Mesh size (m)	Element type	Analysis method
20	3.36	93.43	120	1e-3	0.2	B21	Modal analysis

Table 8.: Random parameters for Monte Carlo simulations of the 2D steel frame structure.

Parameter	Earthquake	Damage location		Stiffness reduction* (%)
		Vertical attribute	Horizontal attribute	
Distribution Value	Uniform [El Salvador,] ElCentro, Chi-Chi]	Uniform 2-combination of [r, c, l]	Uniform 2-combination of [1-5]	uniform [0, 30,60, 90]

*0 : *No damage.*

Table 9.: Label of frame connection.

Connection	Label														
Connection number	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Horizontal attribute	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Vertical attribute	l	l	l	l	l	c	c	c	c	c	r	r	r	r	r

Dynamic analysis is performed with the help of the mode-based dynamic analysis (Awkar & Lui, 1999) and modeling transfer technique in which the outputs of

the first stage are considered as inputs of the second stage. More details about seismic analyses of a frame structure via mode-based dynamic analysis could be found in (Dassault, 2016b). In engineering practice, using linear dynamic analysis method still a preferable choice compared to nonlinear counterpart thanks to its practicality, accuracy of results can be adjusted further via a safety factor. Tables 7 and 8 encloses numerical parameters of the finite element model.

The same procedure presented for the first example is applied for this 2D frame to build deep learning models: performing Monte Carlo simulation, involving parameters' variation, realization of extensive finite element analysis, labelling data by structural states, training deep learning models for each sensors, combining output of all sensors into a final classification result and evaluation of the models' performance. As steel is a homogeneous material and usually manufactured with a highly controlled process, then their properties are more reliable than those of concrete, which is heterogeneous material and can be manually cast in situ. Therefore, one excludes the variation of steel properties from the Monte Carlo simulation in this case study. The main uncertainty sources for Monte Carlo simulations of this example are degrees of stiffness reduction, damage locations, and earthquake records drawn from uniform distributions as listed in Table 8.

For this case study, ones investigate the double damage localization tasks. The multiple damage detection is much more challenging than single damage detection due to the increasing number of scenarios, especially for complex structures. There are 15 frame connections in total; then, the number of scenarios is only 15 for single damage, but rises to 105 cases for double damage, will be 455 cases for triple damage scenarios, and so forth. With a given database, when the number of classes increases, the training sub-dataset for each class will be fewer, thus the overall accuracy decreases. To overcome this limitation, the damage scenarios are divided into different subcategories of a smaller number of classes by incorporating more complementary attributes.

For the considered scenario, further information, i.e., the story number (horizontal attribute), and the related frame columns (vertical attribute), are integrated into the frame connection information apart from node number (see Table 9). There are 5 stories; thus, the number of double damage scenarios with respect to the horizontal attribute reduces to 15 cases. The label for each case is denoted by corresponding stories; for example, label (3, 5) indicates damages occurring in the beams of the third and fifth stories. Whereas with 3 groups of columns: left, center, and right column (denoted by l, c, r), there are only 9 possible cases which are annotated by a pair of numbers, for instance, (l, c) means damage happens in left and center columns. Then, a two-head deep neural network is trained, one head for detecting the story of damaged connections, another for their vertical attribute. By combining these two results, the damaged connections are accurately localized. This multi-head strategy is more amenable in practice than the direct multi-classification approach when dealing with a large number of output classes and unchanged database volumes.

To visualize the classification results, t-Distributed Stochastic Neighbor Embedding (t-SNE) technique is adopted due to its ability of dimensionality reduction in visualizing high-dimensional datasets (Maaten & Hinton, 2008). Since deep learning models will extract multiple patterns from raw data, thus using t-SNE provides a graphical presentation about how efficient each algorithm in clustering different damage scenarios. If input signals of the same damage class are well grouped and separated from other classes, the DL algorithm is effective. In contrast, if data points of different damage scenarios mix together or data points belonging to the same damage class but are not properly grouped, the utilized algorithm needs to be improved.

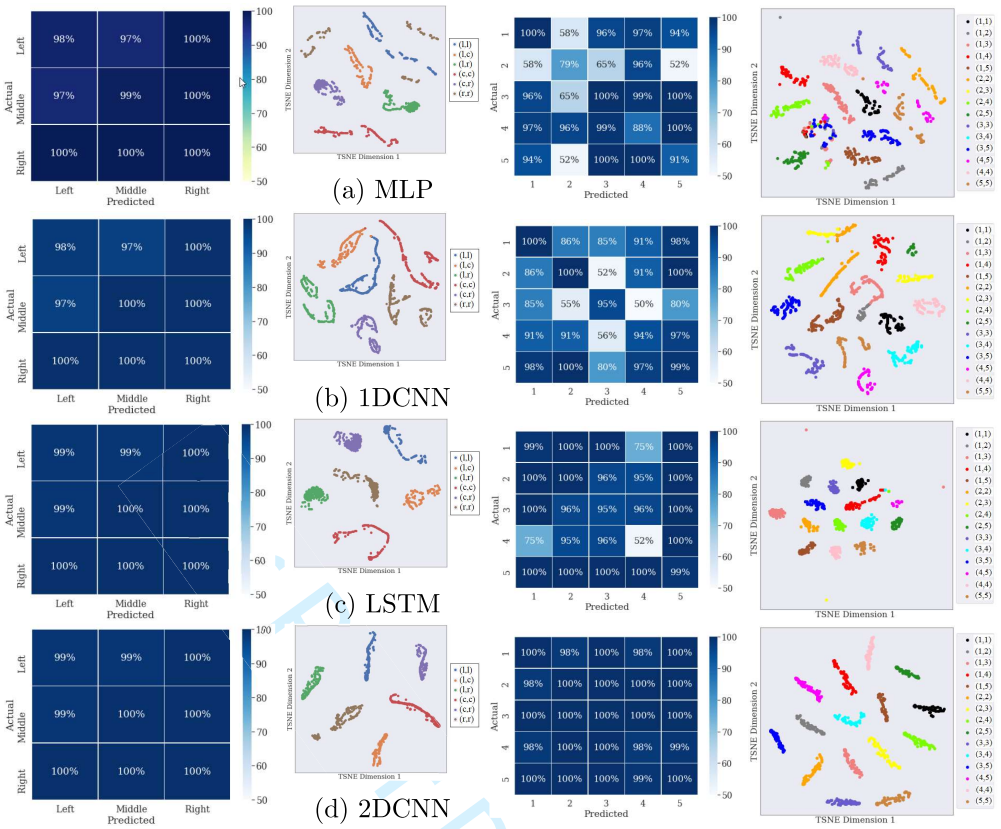


Figure 17.: Damage localization accuracy for double damaged scenarios obtained by different Deep Learning algorithms and the corresponding tSNE visualizations.

Figure 17 demonstrates the results of the stiffness reduction identification for steel frame under earthquake ground motion. The rows of the figure correspond to MLP, LSTM, 1DCNN, and 2DCNN algorithms, respectively. The left side of each row presents the results of detecting the building storey of damaged connection pairs, including the percentage accuracy and their tSNE representation. The right-hand side highlights results related to the vertical attribute. For vertical attributes, all four deep learning algorithms show high accuracies of more than 97%. The tSNE indicates that obtained results for different groups are well separated, but MLP is not as good as the three other algorithms in clustering data, for example, the data of the pair (r,r) are subdivided into three locations. For horizontal attributes of damage localization, as the number of output classes increases to 15, the superiority of LSTM and 2DCNN is well demonstrated. Their average accuracies are 97% and 99% respectively, whereas, for MLP and 1DCNN algorithms, some pairs of damaged connections obtain under 60%, (such as the pair (2, 5) for MLP and pair (3,4) for 1DCNN). The tSNE also highlights this observation. The tSNE of MLP's results shows an area where data points of several classes overlap together at the bottom left corner. In the tSNE of 1DCNN, the pair (2, 5) and (3, 4) are not well split. In contrast, the tSNE of LSTM displays good clustering results, but there are some data points far apart from the others. The tSNE of 2DCNN has the best clustering image.

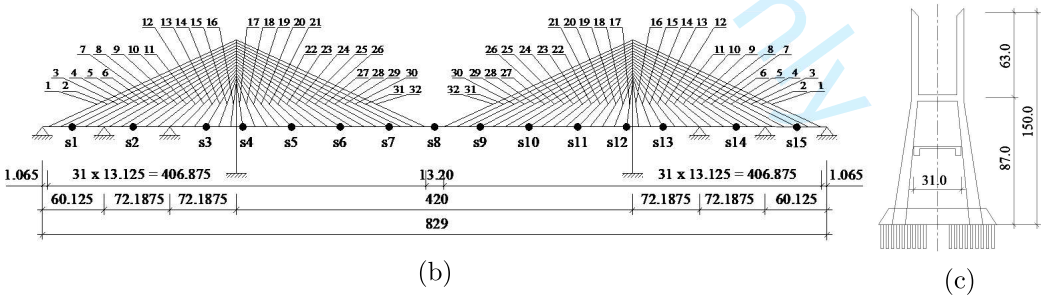
Remark 3: The proposed DL workflow has the ability to diagnose the multi-damage scenarios across the structure with high accuracy. 2DCNN consistently provides the

best detection results, followed by LSTM and 1DCNN. In contrast, the performance of MLP algorithm needs to be improved further by modifying the network's hyperparameters. Theoretically, MLP architecture can attain an accuracy as high as that of 2DCNN, but the hyperparameter optimization is another laborious task, and change radically for different SDD tasks, thus it is more practical to invoke other DL algorithms.

3.4. Case Study 3: 3D Bridge Under Moving Load

For the third case study, the DL approaches are applied to a full-scale bridge structure. In stayed-cable bridges, cable is one of the most crucial components (Montoya, Deodatis, Betti, & Waisman, 2015), having a great impact on the dynamic behavior of the structure. If there is a loss in the prestressed force in cables, the structural rigidity will be significantly reduced. The common method to examine the operational condition of cables is measuring vibrations of each cable, calculating its eigenfrequency values then deriving the actual prestressed force in cables from empirical formulae. However, this method does not allow for directly assessing the interaction between global dynamic responses of the structure and the cable system. Alternatively, by using the presented deep learning methods, one attempts to detect the location of cable prone to prestressed force reduction via bridge vibration time series, which can be obtained from accelerometer sensors installed across the bridge.

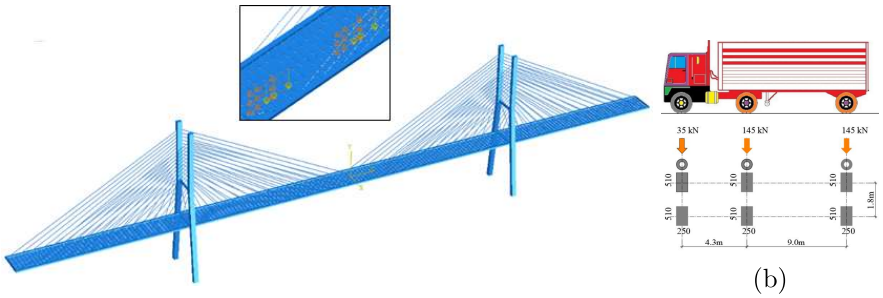
(a)



(b) (c)

Figure 18.: a) Vasco da Gama bridge located in Lisbon, Portugal (*Vasco de Gama bridge*, 2011) b) Schematic representation of the bridge and measurement points' location c) Pylon elevation with dimension in place.

The presented methodologies are applied to the Vasco da Gama cable-stayed bridge



(a) (b)

Figure 19.: a) Finite element model of the Vasco da Gama bridge built in Abaqus and b) a standard vehicular load (HS20 truck) according to American Association of State Highway and Transportation Officials (AASHTO, 1998).

in Lisbon, Portugal (Figure 18(a)), for assessing tension reductions in cables based on dynamic responses of the structure to crossing vehicles. The main structure of the bridge consists of a central span of 420 m length, and three lateral spans of 62, 70.7 and 72 m either side, leading to a total length of 829 m (Pedro, Oliveira, & Reis, 2010). The bridge deck is 31m wide, made by a 25 cm thick concrete slab supported by two longitudinal prestressed girders of 2.6 m depth and I-shaped transverse steel beams spaced at 4.4 m intervals. The structure is suspended by a system of cables composed of 48 stays connected to either side of two outstanding pylons. The pylon has the H shape with a height of 150 m, including two legs whose cross-section dimension varies gradually from 12x7.7 m at the base to 5.5x4.7 m at the top, linked by a transverse beam at the 87 m level. The details of the geometric dimensions of the bridge are illustrated in Figure 18(b, c). In terms of cable parameters, the cross-section of each cable is 0.019 m^2, and an average value of the effective cable tension is selected as 6256 kN after measurements of (Cruz & Miranda, 2017). The material parameters of the bridges are as follows: concrete of C45/55 grade with E_c=44.17 GPa, volumic density of 2400 kg/m^3; structural steel S355J2G3 with E_s=210 GPa and volumic density of 7850 kg/m^3. For the structural damping of the Vasco da Gama Bridge, it is assumed that a damping ratio of 0.3% is applied for all vibration modes.

Table 10.: Numerical input parameters for the 3D stayed-cable bridge case study.

Duration of simulation (s)	Time step (s)	Mesh size (m)	Beam ele. type	Cable ele. type	Deck ele. type	Connection type	Analysis method
60	$1e^{-4}$	1.0	B31	T3D2	S4R	Zero-length connector	Explicit analysis

Table 11.: Random parameters for Monte Carlo simulations of the stayed-cable bridge structure.

Parameter	Number of trucks	Velocity (m/s)	Tension loss (%)	Damage location
Distribution	Uniform	Uniform	Uniform	uniform
Value	[1-5]	[10, 15, 20, 25, 30]	[0, 30, 60, 90]	[0-128]

*0 : *No damage.*

A finite element model of the bridge is developed to simulate the structural behavior when a vehicle travels across, (see Figure 19a) using Abaqus software. The bridge deck is modeled by four noded S4R shell elements from the Abaqus element library. The pylons of the cable-stayed bridges are modeled using 3D beam elements. The element has six degrees of freedom (dofs) at each node, including three translations and three rotations around the x, y and z directions. The stay cables are modeled using two-noded T3D2 truss elements (Olamigoke, 2018) subjected to pre-stressed action. The pre-stressed value per strand is obtained by dividing the tension force by the strand area. In FEM, the tension is modeled as an initial condition, directly applied on cable elements at the beginning of the analysis using predefined field, mechanical category, where the type is selected as stress. The pre-stressed value is the same for all cables except for the damaged one, whose value is virtually reduced by a prescribed damage level. Table 10 lists numerical input parameters for the 3D FEM of the stayed-cable bridge in Abaqus.

In terms of vehicular load, a standard HS20 truck, according to the American Association of State Highway and Transportation Officials (AASHTO, 1998) specifications, is selected in this study. The truck weighs 36 tons, has three axles with configuration and loading details highlighted as in Figure 19(b). Multiple cases of vehicular loads are considered in the calculation, such as one or multiples vehicles traveling at different velocities. Each truck is modeled by 3 concentrated loads of 35, 145 and 145 kN, respectively, moving at a constant velocity on the centerline of the bridge deck. During movement, the distance between these three loads (4.3 m and 9.0 m) remains constant. In the Abaqus software, the subroutine DLoad is a convenient tool to simulate such moving loads. The velocity of trucks is selected from the range [10, 15, 20, 25, 30] m/s, the number of trucks varies from 1 to 5, and the distance between two consecutive vehicles is fixed at 100 m. The structural response is virtually measured at 30 different points (15 points each side) along the deck, as highlighted in Figure 18(b).

Once a detailed 3D FEM of the bridge is constructed, various damage scenarios are introduced into the FEM to generate a synthetic database of structure response. Following the experimental program conducted by Nazarian, Ansari, Zhang, and Taylor (2016), methods using the bridge deck's responses were inefficient for damages of below 30% tension loss in one stay-cable in detecting either damage location or damage severity. Therefore, three levels of tension reduction investigated are 30%, 60%, and 100% (total loss of tension). Furthermore, when working with added-noise data, preliminary calculation with low-level tension loss (about 5%) did not yield satisfying results then was not reported in this manuscript. The damage is arbitrarily introduced to one of 128 cables, at the beginning of the simulation. In total, 10000 simulations are realized, including different damage scenarios, and various vehicular load cases, as summarized in Table 11. The learning process is conducted in a supervised fashion in which each series of dynamic responses of the bridge is labeled by levels of damage or location of damaged cables. The number of samples is uniformly distributed among classes, i.e., approximately 80 samples (10000/128) for damage localization, and 2500 (10000/4) for damage severity.

Figure 20 presents the validation accuracy and training loss retrieved during training processes for two tasks: damage severity and damage localization. The accuracy on the valid dataset is increased as a function of the number of epochs, and the training loss decreases at the same time. Once the valid accuracy becomes stable, and the training process could be terminated. It can be seen that for damage severity tasks, 2DCNN outperforms the other algorithms with the converged accuracy of around 94%, followed by LSTM, 1DCNN, and MLP with accuracy of 90%, 85%, and 80%,

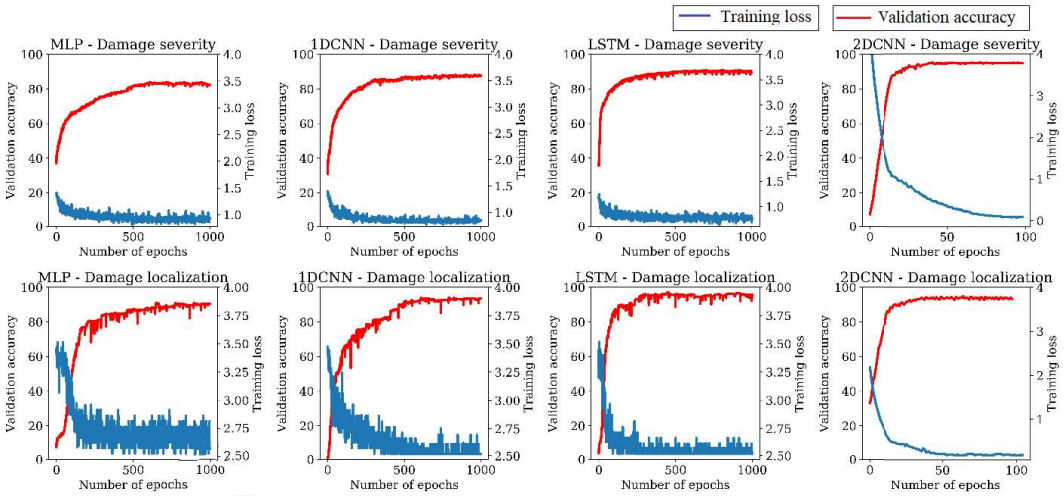


Figure 20.: Evolution of training loss and validation accuracy in function of the number of epochs in the training process.

respectively. It is also noteworthy that the learning curve of 2DCNN becomes stable after only 20 epochs, while it requires more than 200 epochs for other methods. The similar patterns are observed for damage localization tasks. Otherwise, the damage localization tasks receive higher accuracy than the damage severity tasks. A possible explanation is that the influence of each cable on the dynamic behavior of the structure is not the same, a minor tension loss of a critical cable (long stay-cables connected to the middle of the bridge) could cause damage equivalent to a medium tension loss of another cable (short stay-cables close to the pylon).

Table 12.: Stayed-cable bridge's damage detection results obtained with different Deep learning algorithms.

Task	Damage severity				Damage localization			
Model	MLP	1DCNN	LSTM	2DCNN	MLP	1DCNN	LSTM	2DCNN
Storage (Gb)	9.2	9.2	9.2	18.9	9.2	9.2	9.2	18.9
Time for 1 epoch (s)	5.1	11.1	24.9	354.0	4.7	9.5	10.6	290
Total traing time (s)	2550	5550	12450	17700	2350	4750	5300	14500
Testing accuracy	78.8%	87.5%	93.0%	98.9%	83.9%	86.0%	95.0 %	99.2%

The results are obtained after training with 500 epochs for MLP, 1DCNN, LSTM, and 50 epochs for 2DCNN..

Table 12 summarizes the computed results of four Deep learning algorithms for the damage detection of the stayed-cable bridge. Although 2DCNN requires the minimal number of epochs during training, the training time for each epoch (354 s) is significantly higher than that of other algorithms. Therefore, the total training time of 2DCNN is 1.4, 3.2, and 6.9 times higher than that of LSTM (17700 s versus 12450 s), 1DCNN (5550 s), and MLP (2550 s). In addition, apart from the original time-series signals, 2DCNN needs to store associated images of 2D time-frequency representation of vibration signals, therefore its required storage is twice as much. This drawback impedes the practicality of 2DCNN. On the other hand, LSTM, which achieves significantly higher accuracy in a commensurable running time without additional storage, is promising solution in real-time SHM applications. 1DCNN also bears potential for wider use due to its advantages as demonstrated in these case studies: higher accurate

than MLP, very rapid, and no need for extended storage.

3.5. Damage Identification with Noisy Data

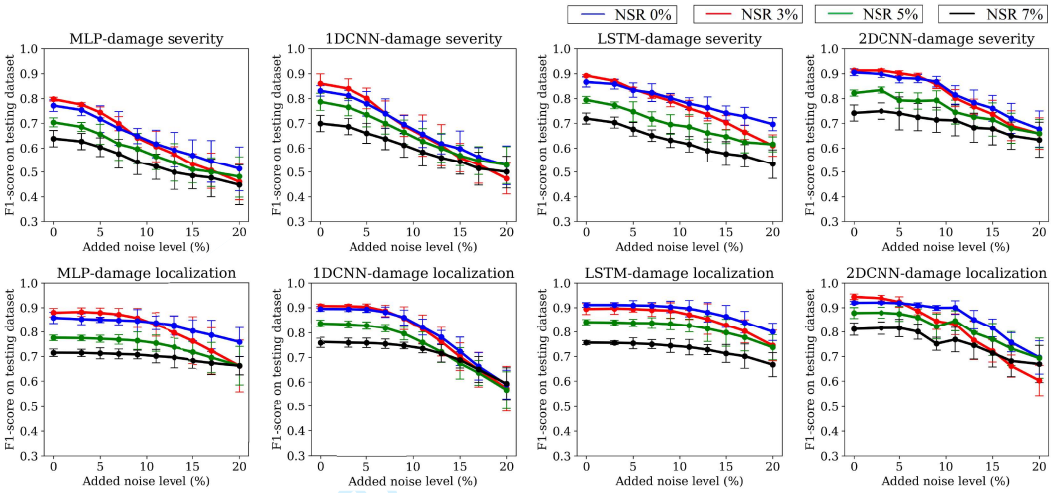


Figure 21.: Evolution of prediction accuracy versus the added-noise level for different Noise-to-Signal ratios.

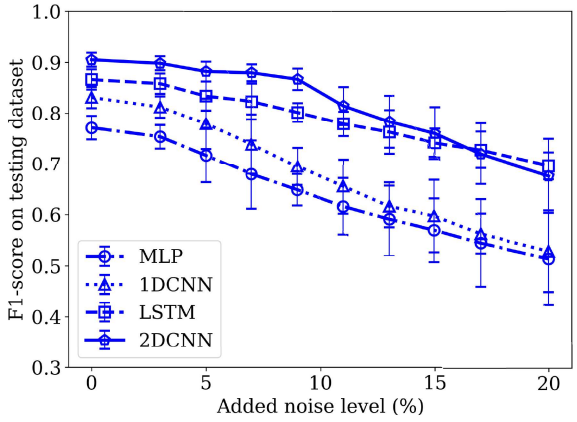


Figure 22.: Detection accuracy on noisy testing data when adding 3% NSR noise to training data.

In the following part, the robustness of the DL is investigated with respect to time-history sensory data contaminated by noise. Noise can be attributed to various sources: one rooted from internal structures' properties, other due to external factors such as device instability, environmental effects, systematic errors. Herein, we consider white-noise, which is a classical yet essential problem for time series analysis (Z. Li, Lam, Yao, & Yao, 2019), reflecting unknown ambient excitation, usually addressed in the SHM literature (de Castro, Baptista, & Ciampa, 2019). Moreover, present methods will automatically learn relevant patterns and mitigate the effect of noise through their hierarchical architecture rather than resorting to a per-noise preprocessing step;

thus, it is potentially applicable to other types of perturbations. Herein, we consider white-noise where its amplitude is defined using root mean square (RMS) value of the vibration data and is represented as follows:

$$x_i^{noise} = x_i + \alpha.\eta_i, \tag{19}$$

where x_i is the vibration signal retrieved from sensor i_t, x_i^{noise} is added-noise data; η_i is the white noise vector with zero mean and unit variance, α is the noise amplitude depended on the RMS of original data.

In the Machine Learning literature, to increase the robustness of models when dealing with noisy data (Yun, Yi, & Bahng, 2001) suggested to train models with low-level added noise dataset instead of virgin datasets, referred as the noise injection learning method. Thereby, the models are retrained with training data augmented by noise-to-signal ratios (NSR) 3%, 5%, and 7% then tested with data corrupted by noise with NSR up to 20%.

Figures 21 and 22 illustrate the relationship of damage detection accuracy with noisy data for investigated methods. In Figure 21, the first and second rows correspond to the damage severity task and damage localization as described above. In each subfigure, there are four curves represent four training dataset augmented with 0%, 3%, 5%, and 7% noise, plotted in blue, red, green and black, the y-axis denotes damage detection accuracy in percentage, the x-axis indicates testing data corrupted by noise up to 20% NSR. For better observation, Figure 22, directly compares the testing performance of four neural network-based methods trained with 3% augmented training data on damage severity tasks.

In addition, to estimate the uncertainty of testing output, each calculation is repeated 10 times, i.e., investigated noises are regenerated and reintegrated into the database, then respective standard deviations are derived. Apparently, the errors, as well as the variation of prediction increases as NSR becomes more pronounced. Among four DL algorithms, 2DCNN shows the longest plateau regions up to around 10% noise level, which means the algorithm could maintain its high performance despite added-noise. It is also noteworthy that the decreasing rate in the performance of LSTM with respect to the noise is relatively low compared to others. Thus, it is recommended that 10% SNR is the upper bound of noise for algorithms investigated in this study; otherwise, the accuracy decreases to below 80%. Moreover, the influence of added noise on the algorithms' performance in damage severity tasks is more pronounced than those in damage localization, as the performance curves of the former decrease more significantly with increasing noise levels.

In terms of injection learning, it is noted that DL algorithms trained on low-level added noise database (Figure 21), i.e., 3% (red curve) initially achieve higher scores compared to noiseless data (green curve), but then rapidly decay with increasing SNR (larger than 10%). On the other hand, training data with 7% SNR noise (black curve) is not beneficial for the model performance, even when testing with non-perturbated data, accuracies around 70% are observed; however, it is still informative as the respective decay rate is lower than that of 0% or 3% SNR noise. A possible explanation for this phenomenon is that once trained with 3% NSR data, the model will perform well with NSR around 3%, but over-noisy data could deteriorate underlying features learned by models and also introduce new fictive patterns. Eventually, patterns of noise will become dominating, thus significantly offsetting the performance of models trained with free and low-noise data. Another likely reason is that one opts the same training process setting for all NSR levels, i.e., the same number of epochs, learning

rate, batch size, which are optimal for noise-free data but not 3% SNR case.

Remark 4: 2DCNN consistently provides the highest performance; moreover, 2DCNN and LSTM retain better performance when dealing with noisy data. Though all investigated methods yield satisfying results with noises of no more than 10% NSR, there exist, in reality, other types of perturbation with potentially higher levels caused by stochastic vehicle loading, temperature effect, sensor instability to address. After all, Deep learning methods directly using time-series are alternative and also complementary SDD approaches, providing more insight into structural behaviors.

4. Conclusions

In this article, four prominent deep learning algorithms that achieved state-of-the-art results in a spectrum of applications are extended for structural damage detection. In order to illustrate steps of realization and compare the performances of these algorithms, four case studies subjected to various external excitation were conducted. Based on the calculation results, some concluding remarks can be given as follows:

- In general, building a neural network-based model for structural damage detection includes the following main steps: first, preparation of a database labeled by respective structural states of concerns, which can be obtained through extensive measurement or Monte Carlo simulations. Second, a neural network architecture is selected and adapted based on given databases and SHM tasks. Third, the training and validation are proceeded to determine optimal parameters of the neural network by using an optimization algorithm (Adam) and gradient computation algorithm (Backpropagation). The performance of the model is then estimated through statistics of output results such as true positive, false positive, and F1 score, etc.
- In comparison to other approaches, neural network-based models are alternative and complementary methods directly using measured vibrational signals without requiring an additional step to extract structural characteristics such as modal identification. Moreover, it is flexible to conduct different damage detection tasks with the same neural network architecture but the last output layer to be fine-tuned per task. Once the models are trained with appropriate datasets and their parameters are stably determined, they could deliver monitoring assessment in a near real-time fashion due to the fast inference time.
- Through various case studies, it is demonstrated that present methods are able to perform not only the SHM level 1 task, i.e., damage detection but also SHM level 2, namely damage localization and damage severity with high accuracy thanks to the feature learning capability of the neural networks and a proposed flexible modular working flow combining patterns from multiple vibration signals. In addition, when dealing with a high number of structural states, a procedure dividing original states into combinations of sub-categories and fine-tuning the network into multi-head, each head for a sub-category, is advisable to achieve satisfying results.
- When comparing four neural networks investigated in this work, it is noted that 2DCNN method constantly provides the best identification results, while LSTM and 1DCNN methods balance the performance and the complexity, i.e., fast in the training process, easy to modify for new structural problems, and does not require storage for images like 2DCNN. And, MLP algorithm has the most

straightforward architecture and low computational complexity; therefore, it is suitable to serve as a baseline model for any new structures before designing a more sophisticated approach.

- A robustness study with noisy data is carried out, showing that the investigated methods can maintain a sufficiently good performance with data corrupted by noise up to 10% NSR. However, with a higher noise level, the performance degraded to some extent.

Therefore, improving the robustness of SHM neural network-based methods with respect to noises of considerably high levels (> 10% NSR) is necessary for future research. It would also be interesting to incorporate probabilistic frameworks into neural networks, for example, Bayesian theory, in order to empower the performance of algorithms, providing not only point estimates but also according confidence interval in response to the unpredictability of noisy input. Another research direction is to fuse various types of data apart from vibration such as strain measurement, temperature variation, material properties' changing pattern, etc. for detecting more range of defects in the structure.

Acknowledgments

This work was supported by the Newton Fund Institutional Links through the U.K. Department of Business, Energy, and Industrial Strategy and managed by the British Council under Grant 429715093.

Conflict of interest

The authors declare no potential conflict of interests.

References

AASHTO. (1998). *Aashto lrfd bridge design specifications.* American Association of State Highway and Transportation Officials.

Abdeljaber, O., Avci, O., Kiranyaz, M. S., Boashash, B., Sodano, H., & Inman, D. J. (2018). 1-d cnns for structural damage detection: Verification on a structural health monitoring benchmark data. *Neurocomputing*, *275*, 1308-1317.

Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., & Inman, D. J. (2018). Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks. *Journal of Sound Vibration*, *424*, 158-172.

Awkar, J. C., & Lui, E. M. (1999). Seismic analysis and response of multistory semirigid frames. *Engineering Structures*, *21*(5), 425–441.

Brownlee, J. (2019). *What is deep learning?* Retrieved from `https://machinelearningmastery.com/what-is-deep-learning`

Burkov, A. (2019). *The hundred-page machine learning book.* Andriy Burkov Quebec City, Can.

Cruz, C., & Miranda, E. (2017). Dynamic tests on large cable-stayed bridge. *Engineering Structures*, *138*, 324 - 3366.

Csébfalvi, A. (2007). Optimal design of frame structures with semi-rigid joints. *Periodica Polytechnica Civil Engineering*, *51*(1), 9-15.

Dassault, S. (2016a). *Abaqus analysis user's manual.*

Dassault, S. (2016b). *Abaqus example problems manual.*
Dassault, S. (2016c). *Abaqus theory guide.*
de Castro, B. A., Baptista, F. G., & Ciampa, F. (2019). New signal processing approach for structural health monitoring in noisy environments based on impedance measurements. *Measurement*, *137*, 155-167.
Fragiacomo, M., Amadio, C., & Macorini, L. (2004). Seismic response of steel frames under repeated earthquake ground motions. *Engineering Structures*, *26*, 2021-2035.
François, C. (2017). *Deep learning with python.* Manning Publications Company.
Galloway, G. S., Catterson, V. M., Fay, T., Robb, A., & Love, C. (2016). Diagnosis of tidal turbine vibration data through deep neural networks. In *Proceedings of the Third European Conference of the Prognostics and Health Management Society.*
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT press.
Haddadi, H., Shakal, A., Stephens, C., Savage, W., Huang, M., Leith, W., & Borcherdt, R. (2008). Center for engineering strong-motion data (cesmd). In *Proceedings of the Fourteenth World Conference on Earthquake Engineering.*
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee Conference on Computer Vision and Pattern Recognition.*
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.
Howard, J., & Gugger, S. (2020). Fastai: A layered api for deep learning. *Information*, *11*(2), 108.
Ince, T. (2019). Seismic response of steel frames under repeated earthquake ground motions. *Electrical Engineering*, *101*(2), 599-608.
Janocha, K., & Czarnecki, W. M. (2017). *On loss functions for deep neural networks in classification.* (preprint arXiv:1702.05659)
Jensen, S. K., Pedersen, T. B., & Thomsen, C. (2017). Time series management systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *29*(11), 2581-2600.
Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*, 1412.6980.
Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2019). *1d convolutional neural networks and applications: A survey.* (preprint arXiv:1905.03554)
Kosmatka, S. H., Kerkhoff, B., & Panarese, W. C. (2019). *Design and control of concrete mixtures.* Portland Cement Association Skokie, IL.
Lee, F. F., Karpathy, A., & Johnson, J. (2018). *Cs231n: Convolutional neural networks for visual recognition.* Retrieved from `http://cs231n.stanford.edu`
Lei, J., Liu, C., & Jiang, D. (2019). Fault diagnosis of wind turbine based on long short-term memory networks. *Renewable Energy*, *133*, 422-432.
Li, J., Dackermann, U., Xu, Y. L., & Samali, B. (2011). Damage identification in civil engineering structures utilizing pca-compressed residual frequency response functions and neural network ensembles. *Structural Control Health Monitoring*, *18*(2), 207-226.
Li, Z., Lam, C., Yao, J., & Yao, Q. (2019). On testing for high-dimensional white noise. *The Annals of Statistics*, *47*(6), 3382-3412.
Lin, Y. Z., Nie, Z. H., & Ma, H. W. (2017). Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil Infrastructure Engineering*, *32*(12), 1025-1046.
Liu, H., & Zhang, Y. (2020). Bridge condition rating data modeling using deep learning algorithm. *Structure and Infrastructure Engineering*, 1-14.
Los alamos national laboratory. (2002). Retrieved from `http://www.lanl.gov/projects/damage_id/index.htm`
Lu, C., Wang, Z., & Zhou, B. (2017). Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification. *Advanced Engineering Informatics*, *32*, 139-151.
Ma, M., Sun, C., & Chen, X. (14). Deep coupling autoencoder for fault diagnosis with multimodal sensory data. *IEEE Transactions on Industrial Informatics*, *3*, 1137-1145.

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(Nov), 2579-2605.

Montoya, A., Deodatis, G., Betti, R., & Waisman, H. (2015). Physics-based stochastic model to determine the failure load of suspension bridge main cables. *Journal of Computing in Civil Engineering*, *29*(4), B4014002.

Nazarian, E., Ansari, F., Zhang, X., & Taylor, T. (2016). Detection of tension loss in cables of cable-stayed bridges by distributed monitoring of bridge deck strains. *Journal of Structural Engineering*, *142*(6), 04016018.

Noguchi, T., & Nemati, K. M. (1995). Relationship between compressive strength and modulus of elasticity of high strength concrete. *Journal of Structural and Construction Engineering*, *474*(1), 1-10.

Olah, C. (2015). *Understanding lstm networks.* Retrieved from `https://colah.github.io/posts/2015-08-Understanding-LSTMs`

Olamigoke, O. (2018). *Structural response of cable-stayed bridges to cable loss.* (Doctoral dissertation, University of Surrey)

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359.

Pedro, J., Oliveira, J., & Reis, A. J. (2010). Nonlinear analysis of composite steel–concrete cable-stayed bridges. *Engineering Structures*, *32*(9), 2702-2716.

Sekulovic, M., & Salatic, R. (2001). Nonlinear analysis of frames with flexible connections. *Computers & Structures*, *79*(11), 1097-1107.

Tang, Z., Chen, Z., Bao, Y., & Li, H. (2016). Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Structural Control and Health Monitoring*, *26*(11), 2296.

Vasco de gama bridge. (2011). Retrieved from `https://arab-trip.com/wp-content/uploads/2019/07`

Wang, Y. R., P, & Gao, R. (2017). Virtualization and deep recognition for system fault classification. *Journal of Manufacturing Systems*, *44*, 310-316.

Yuan, M., Wu, Y., & Lin, L. (2016). Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *Ieee International Conference on Aircraft Utility Systems.*

Yun, C. B., Yi, J. H., & Bahng, E. Y. (2001). Joint damage assessment of framed structures using a neural networks technique. *Engineering Structures*, *23*(5), 425-435.

Zhang, Y., Miyamori, Y., Mikami, S., & Saito, T. (2019). Vibration-based structural state identification by a 1-dimensional convolutional neural network. *Computer-Aided Civil Infrastructure Engineering*, *34*(9), 822-839.

Zhao, R., Wang, J., Yan, R., & Mao, K. (2016). Machine health monitoring with lstm networks. In *10th International Conference on Sensing Technology.*